

УДК 378:004.032.8

© **Грабовський Є. М.**, к.е.н., доцент, Харківський національний економічний університет імені Семена Кузнеця, Харків, Україна

МЕТОДИКА СТВОРЕННЯ МУЛЬТИМЕДІЙНОГО НАВЧАЛЬНОГО ДОДАТКУ З ВИКОРИСТАННЯМ МОЖЛИВОСТЕЙ РЕДАГУВАННЯ КОНТЕНТУ

Abstract. The technique of developing a multimedia application with visual editing tools is presented in the paper. The main stages of the proposed methodology are specified. The stages of publication in Internet editions are considered.

Будучи ключовою технологією всесвітньої павутини, мова розмітки HTML існує вже стільки, скільки існує Інтернет, і розвивається в міру розвитку Мережі. Для перегляду веб-сайтів використовуються веб-браузери, чії розробники, часто конкуруючи між собою, додають в браузері різні нові можливості та технології. Для того, щоб ці технології розвивалися спільно і підтримувалися основними браузерами, організація під назвою «консорціум всесвітньої павутини» займається розробкою специфікацій мови розмітки. Актуальною на сьогодні версією специфікації є HTML5. У п'ятій версії мови крім ряду нововведень були стандартизовані технології та методики, раніше доступні в деяких з браузерів. Однією з цих технологій є можливість візуального редагування розмітки за допомогою спеціальної властивості contenteditable.

На сьогодні існує єдиний стандарт, розробкою і розвитком специфікацій якого займається організація «Консорціум Всесвітньої Павутини», W3C. Ці специфікації є єдиною існуючою «офіційною документацією» за даними технологіями.

Оскільки документ описує стандарт, специфікації W3C чітко структуровані, скрупульозно і дуже важкі до прочитання. Зважаючи на це вивчення молодим спеціалістом вищезазначених технологій по специфікації представляється скрутним.

Проблема в даний час вирішена численними навчальними посібниками різних авторів, представленими на ринку. Проте варто враховувати, що будучи технологіями, що відповідають за візуальне представлення інформації, HTML, JavaScript і CSS для свого вивчення потребують, насамперед, практики їх застосування, яку нездатні дати навіть кращі з існуючих навчальних посібників. Тому важливого значення набуває створення науково обґрунтовано методики формування навчального додатку з використанням можливостей редагування контенту.

Метою дослідження є розробка методики створення мультимедійного посібника на основі технології HTML5 з елементами contenteditable для отримання учнями навичок візуального редагування вмісту сторінки.

Вперше представлена в браузері Microsoft Internet Explorer версії 5.5, властивість була доступна і раніше, до публікації стандарту HTML5, однак лише в цій специфікації особливості застосування contenteditable були стандартизовані.

За допомогою цієї властивості користувач може редагувати частину сторінки прямо в браузері, в режимі «Отримуємо те, що бачимо» («What You See Is What You Get», WYSIWYG), в режимі реального часу, що дозволяє значно підвищити наочність і знизити часові витрати на підготовку вмісту сторінок.

На рис. 1.представлений приклад використання WYSIWYG з сайту tumblr.com. У такого підходу є ряд переваг перед написанням розмітки вручну:

1) наочність подання. Текст не містить ключових слів і виглядає максимально наближено до кінцевого результату (в окремих випадках текст уже є кінцевим результатом);

2) доступність користувачеві. Редагування тексту в більшості випадків застосування даної властивості взагалі не вимагає від користувача ніяких спеціальних знань. Людина без знання мов розмітки може редагувати вміст за допомогою інтерфейсу, наближеного до інтерфейсу звичних йому текстових процесорів, таких як Microsoft Office Word, LibreOffice Writer і AbiWord.

Ключові технології в світі Інтернет — мова гіпертекстової розмітки HTML, мова сценаріїв JavaScript і каскадні таблиці стилів CSS. Так як сучасне життя важко уявити без Інтернет — ці технології можна сміливо віднести до найбільш затребуваних.

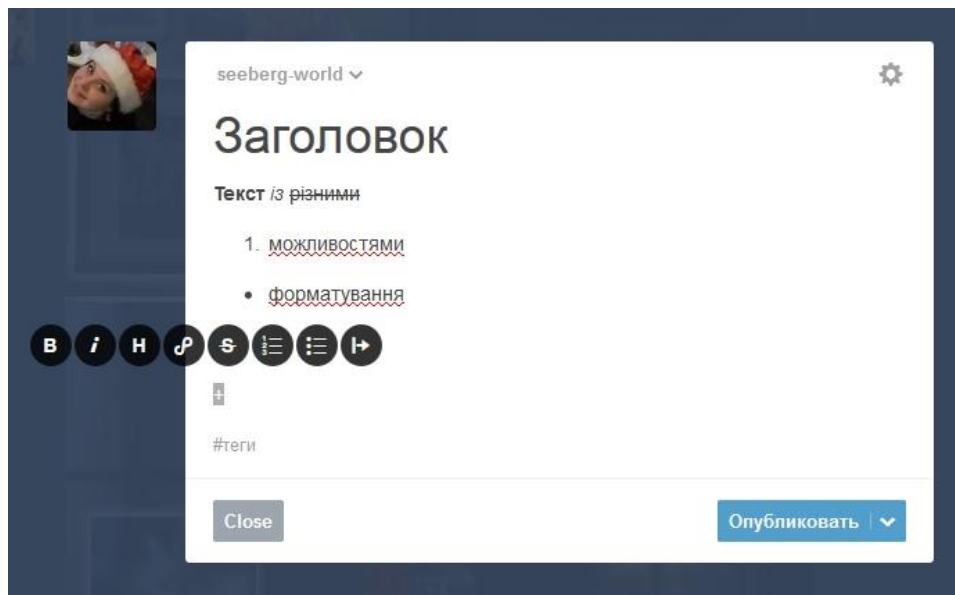


Рис. 1. Приклад використання WYSIWYG

Проаналізуємо зміст типової навчальної програми, присвяченої веб-технологіям. В рамках програми вивчаються такі розділи:

- типи веб-сторінок;
- гіпертекст;
- структура веб-сторінки;
- оформлення тексту;
- редагування готової веб-сторінки;
- гіперпосилання;
- списки;
- вставка малюнків;
- таблиці;
- фрейми;
- організація зв'язків між сторінками;
- побудова веб-сайтів.

Виходячи з даного списку можна зробити висновок, що вивченню веб-технологій приділяється досить мало часу, у зв'язку з чим постає необхідність студентів, бажаючих оволодіти навичками веб-програмування, в самостійному вивченні вищезазначених технологій.

Особливої уваги ця технологія заслуговує в області розробки мультимедійних навчальних програм: наочність і доступність підходу позитивно позначаються на ергономіці сприйняття навчального матеріалу, що позитивно відбивається на навчальній мотивації та ефективності навчання.

У зв'язку з тим, що специфікація HTML5 щодо нова (фінальна версія специфікації була опублікована тільки в жовтні 2014) інформація про методи розробки навчальних мультимедійних додатків з використанням технології розрізнена і безсистемна.

Серед доступних джерел інформації важко виділити такі, які є значущими, крім, власне, самого тексту специфікації.

Розвиток веб-технологій загалом і технології HTML5 (а також її специфікації) — безпосередньо залежить від розвитку веб-браузерів, розробкою яких займаються різні великі корпорації.

Інструменти розробки на базі технології HTML5 не формалізовані, у зв'язку з чим не всі користувачі здатні оволодіти навичками роботи з ними. Крім специфікації стандарту, в мережі існує велика кількість онлайн-ресурсів, присвячених HTML5, серед яких можна виділити такі, як htmlbook.ru, Mozilla Developer Network і проєкт Google HTML5ROCKS. Однак інформація цих ресурсів часто неповна або суперечлива, у зв'язку з чим рекомендується звертатися до офіційної специфікації.

У ході дослідження розглядається процес створення електронних навчальних посібників на базі HTML5 і JavaScript з використанням вищезазначених елементів з `contenteditable`. Такі посібники допоможуть кого навчають краще ознайомитися з технологією HTML5 на наочних прикладах. Розглядаються існуючі методи побудови мультимедійних додатків на базі технології HTML5, проведено аналіз практичного досвіду програмної реалізації існуючих електронних навчальних посібників, визначені слабкі та сильні технологічні боку цих методів.

Навчальні посібники, подібні до того, що розробляється в рамках даного проєкту, безсумнівно, існують. У мережі досить широко поширені онлайн-сервіси, що дозволяють користувачеві знайомитися з веб-технологіями на «живих прикладах». Найпопулярнішими з таких сервісів є англomовний codecademy.com і російськомовний htmlbook.ru.

На основі систематизації інформації літературних джерел [1-6] пропонується здійснення розробки подібних мультимедійних посібників в кілька етапів.

1) Етап 1. Пошук і підготовка тексту навчального матеріалу, який буде додано до електронного видання. Слід враховувати орієнтованість програми на наочну форму навчання — отже уникати великих обсягів тексту. Рекомендується обмежити обсяг тексту між двома інтерактивними елементами двома-трьома абзацами.

2) Етап 2 (опціональний). Доробка програми під конкретний матеріал. Розробка розширень для нових інтерактивних елементів.

3) Етап 3. Імпорт текстової та репрезентативною інформації. На цьому етапі здійснюється інтеграція матеріалу в додаток, переклад тексту в використовуваний додатком формат, настройка інтерактивних елементів.

4) Етап 4 (опціональний). Зміна шаблону оформлення та додавання в додаток нових елементів дизайну.

5) Етап 5. Збірка додатку. На цьому етапі додаток наводиться в готовий до публікації вид. В даний момент (в прототипі) збірка здійснюється вручну, в майбутньому, при необхідності публікації великих обсягів матеріалу, можлива розробка автоматичного збирача.

6) Етап 6. Тестування програми та інтерактивних елементів.

Після підготовки матеріалу додаток можна розмістити «як є» на веб-сайті для подальшого доступу до матеріалу через Інтернет або поширювати на електронних носіях.

Для завантаження даних у додатку слід використовувати технологію JSONP для обходу Cross-Origin Policy браузерів при підвантаження локальних файлів.

Також у додатку для маніпуляції об'єктною моделлю документа (DOM) має використовуватися бібліотека jQuery, що де-факто є стандартом індустрії веб-розробки.

Як уже згадувалося раніше, властивість `contenteditable` у будь-якого HTML-елемента, встановлене як «true», дозволяє користувачеві редагувати текстовий вміст цього елемента, зберігаючи його форматування.

Використання веб-стандартів в розробці додатку дозволяє досягти більшого ступеня портативності кінцевого додатку, тому що веб-технології відіграють важливу роль в житті сучасного суспільства і, з одного боку, постійно розвиваються, пристосовуючись під всі запити користувача, а з іншого боку – підтримуються широким спектром різних операційних систем і пристроїв.

Важливу роль відіграє простота розробки під платформу веб: HTML і CSS легкі в освоєнні і відмінно пристосовані для виведення текстової інформації, а JavaScript є потужною і сучасною мовою сценаріїв з нестрогою типізацією.

Для поширення додатки була обрана найпростіша з можливих форм: набір html і супутніх файлів. Не дивлячись на існування ряду технологій, таких як Electron або NWJS, що дозволяють створювати повноцінні stand-alone додатки, засновані на веб-стандартах, їх використання сповільнило б розробку і ускладнило інтеграцію матеріалу в додаток. Такий підхід також сильно ускладнив би можливу подальшу публікацію заснованих на додатку посібників на веб-сайтах. Має сенс згадати про можливість подальшої модифікації додатку бажаними під свої потреби. Для цього в додатку передбачена можливість створення модулів.

Існує два основних підходи до розробки подібних програм: розробка так званих «односторінкових клієнтських додатків» (SPA, «Single-page applications») та розробка традиційних гіпертекстових медіа.

Ключовою особливістю односторінкових додатків є те, що вони представляють із себе один HTML-файл, що використовує JavaScript для завантаження і обробки інформації з різних джерел, таких як веб-сервіси та різноманітні автоматизовані програмні інтерфейси. Даний підхід дозволяє задати загальне для всього програми оформлення, а також, у разі, якщо додаток представляє із себе веб-сайт, звести до мінімуму використання серверних мов програмування і потенційно знизити навантаження на сервер у зв'язку з відсутністю необхідності попередньої генерації та завантаження частин розмітки, відповідальних за оформлення сторінок, при кожній зміні виду у додатку.

Для створення односторінкових веб-додатків на мові JavaScript в наш час існує велика кількість спеціалізованих технологій і бібліотек, таких як BackboneJS, Angular, EmberJS та інші.

Традиційні гіпертекстові посібники представляють собою безліч HTML-файлів, пов'язаних між собою гіперпосиланнями. Природно, такий підхід жодною мірою не виключає використання серверних мов програмування, що дозволяють генерувати HTML-розмітку «на льоту», використовуючи дані з таблиці в базі даних, однак такі посібники вимагають попередньої настройки веб-сервера і сервера баз даних, внаслідок чого говорити про їх портативності не доводиться. Тому матеріал посібників, як правило, генерують у вигляді безлічі статичних HTML-файлів заздалегідь.

Навчальний посібник, повинен використовувати перший з цих методів, тобто представляє із себе односторінковий веб-додаток, завантажувати JSON-дані з файлу. Такий підхід був обраний тому, що розроблені з його допомогою додатки в майбутньому можна легко модифікувати, додаючи новий вміст (методичний матеріал), причому для цього не потрібне спеціальне ПЗ, а необхідний для цих змін набір знань - мінімальний.

Для створення методичного посібника, що дозволяє користувачеві редагувати це форматування і бачити результат у вигляді вихідного коду (розмітки) використаний готовий, заснований на властивості `contenteditable`, написаний на JavaScript і використовуваний в якості компонента веб-сторінки візуальний редактор з відкритим вихідним кодом TinyMCE.

Фреймворк складається з набору шаблонів і компонентів, таких як готова верстка адаптивних сіток (responsive grid system), що дозволяє створювати front-end частину додатку

з урахуванням майбутньої роботи на мобільних пристроях, система розмітки для типографіки (спеціально пристосовані для легкого читання співвідношення розмірів відступів і кегля), готові JavaScript-елементи (для спливаючих повідомлень, елементів навігації та іншого), готова система розмітки таблиць, оформлення форм (в тому числі – інтерактивні JavaScript-компоненти), стилі для різного роду макетів і міток, а також набір готових модулів розмітки, дозволяють в стислі терміни створити життєздатний прототип веб-додатки. Також bootstrap включає в себе спеціальний шрифт з іконками, що дозволяє вбудовувати в додатки векторні символи (всього їх 250). Шрифт підтримується всіма сучасними веб-браузерами і добре виглядає навіть на мобільних пристроях (в тому числі на мобільних пристроях з високою роздільною здатністю екрану).

Фреймворк включає в себе шаблони оформлення сіток, макетів, форм, міток, типографіки, спливаючих повідомлень, таблиць, елементів навігації та іншого.

Основні інструменти фреймворку включають в себе:

- 1) сітки — наперед задані розміри колонок і обгортки для верстки;
- 2) обгортки — дозволяють визначити фіксовану або «гумову» верстку;
- 3) типографіку — наперед задані розміри шрифтів і відступів, визначення деяких блоків тексту, таких як код, цитати і так далі;
- 4) медіа — елементи, що дозволяють спростити вбудовування аудіо і відео на сторінку;
- 5) таблиці — засоби оформлення та сортування таблиць;
- 6) форми — інструменти для оформлення форм і їх подій, включаючи валідацію і автозаміну;
- 7) елементи навігації — посторінкова навігація, «хлібні крихти», меню навігації і т.д.;
- 8) спливаючі вікна — модальні та немодальні вікна сповіщень.

Для відображення вихідного коду програми слід застосовувати редактор коду CodeMirror, що представляє із себе JavaScript-компонент текстового поля з підсвічуванням вихідного коду.

Запопновані етапи методики дозволяють створити додаток простим і доступним способом, що дозволяє в стислі терміни самостійно реалізувати мультимедійні посібники для електронних видань.

Перелік посилань:

1. Aralova, N. (2017). The method of technology evaluation based on improved cost approach. *Science and Innovation*, 13(3), 65-76 Doi: <https://doi.org/10.15407/scine13.03.065>
2. Martins, P. (2017) A Web-based Tool for Business Process Improvement. *International Journal of Web Portals*, Volume 9, Issue 1, 68 – 84 DOI: <https://doi.org/10.4018/IJWP.2017070104>
3. Hryshchuk, R. (2017) Synergetic control of social networking services actors' interactions. *Recent Advances in Systems, Control and Information Technology*, 543, 34-42. DOI: https://doi.org/10.1007/978-3-319-48923-0_5.
4. Hopp, T., Gangadharbatla, H. (2016) Novelty Effects in Augmented Reality Advertising Environments: The Influence of Exposure Time and Self-Efficacy. *Journal of Current Issues & Research in Advertising*, 37 (2), 113–130. DOI: <https://doi.org/10.1080/10641734.2016.1171179>
5. Brambilla, M. (2014) Large-scale Model-Driven Engineering of web user interaction: The WebML and WebRatio experience. *Science of Computer Programming*, Volume 89, Part B, 71-87 DOI: <https://doi.org/10.1016/j.scico.2013.03.010>
6. Chen, L., Ellis, S., Suresh, N. (2016) A supplier development adoption framework using expectancy theory. *International Journal of Operations and Production Management*, 36(5), 592-615. DOI: <https://doi.org/10.1108/IJOPM-09-2013-0413>.